

Image Segmentation and Multiple skew estimation, correction in printed and handwritten documents

*Thesis submitted in
partial fulfillment of the requirements
for the degree of*

Bachelor of Technology

in

Computer Science and Engineering

by

Sai Prasanth Gumpalli

110CS0479

Prasanth Kandipalli

110CS0141

under the guidance of

Prof. Ramesh Kumar Mohapatra



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India
May 2014



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Prof. Ramesh Kumar Mohapatra
Assistant Professor

May, 2014

Certificate

This is to certify that the work in the thesis entitled *Image Segmentation and Multiple skew estimation, correction in printed and handwritten documents* by *Sai Prasanth Gumpalli and Prasanth Kandipalli* is a record of an original research work carried out under my supervision and guidance in partial fulfilment of the requirement for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Ramesh Kumar Mohapatra

Acknowledgement

This dissertation, though an individual work, has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough.

We would like to gratefully acknowledge the enthusiastic supervision and guidance of Prof. Ramesh Kumar Mohapatra during this work. His suggestions and constant encouragement proved to be an immense source of motivation.

We are very much indebted to Prof. Banshidhar Majhi, for guiding us throughout the project. Also for the resources and facilities that were available to us whenever we needed, that proved to be a vital part of the success of this work.

Our sincere thanks to all of the professors for being our knowledge resource.

Last but not the least, we are indebted to NIT Rourkela for providing us such a platform where we learned a lot and gained so much experience.

Sai Prasanth Gumpalli
Prasanth Kandipalli

Authors Declaration

We hereby declare that all the work contained in this report is our own work unless otherwise acknowledged. Also, all of our work has not been previously submitted for any academic degree. All sources of quoted information have been acknowledged by means of appropriate references.

Sai Prasanth Gumpalli
Prasanth Kandipalli

NIT Rourkela

Abstract

Analysis of handwritten document has always been a challenging task in the field of image processing. Various algorithms have been developed in finding solution to this problem. The algorithms implemented here for segmentation and skew detection works not only on printed or scanned document images but for also handwritten document images which creates an edge over other methodologies.

Here Line segmentation for both printed and handwritten document image is done using two methods namely Histogram projections and Hough Transform assuming that input document image consists of no major skews. For Histogram Projection to work correct, the document must not contain even slight skews. Hough transform gives better results than the former case.

Word Segmentation can be done using the connected components analysis. Here, we first identify connected components in the printed or handwritten document image. A methodology is being used here which detects multiple skews in multi handwritten documents or printed ones. Using clustering algorithms, we detect multiple skew blocks in a handwritten document image or printed document image or a combination of both. The algorithm used here also works for skewed multi handwritten text blocks.

Contents

Certificate	ii
Acknowledgement	iii
Authors Declaration	iv
Abstract	iv
List of Figures	viii
1 Introduction	1
1.1 Digital Image Processing in Documents	1
1.2 Problem Definition	1
1.3 Organization of thesis	2
2 Segmentation	3
2.1 Line Segmentation	3
2.1.1 Using Histogram Projection	3
2.1.2 Using Hough Transform	5
2.2 Word Segmentation	9
2.2.1 Connected Component Analysis	9
3 Skew Detection and Correction in single skewed printed or handwritten documents	11
3.1 Required Morphological Operations	12
3.2 Flow Chart of the method	13
3.3 Results	14

4	Multi Skew Estimation and Correction in multi handwritten or printed documents	15
4.1	Required Morphological Operations	15
4.2	Clustering Analysis	16
4.3	Selection of Clustering Algorithms	19
4.4	Flow Chart of the method	20
4.5	Results	21
5	Conclusion and Future Work	25
5.1	Achievements and Limitations	25
5.2	Further Developments	25
	Bibliography	26

List of Figures

2.1	Input Handwritten document	4
2.2	Horizontally segmented using Histogram Projection	5
2.3	Segmented using Hough Transform (with low threshold value)	8
2.4	Segmented using Hough Transform (with high threshold value)	8
2.5	Input handwritten document	10
2.6	Word Segmented Output	10
3.1	Image after applying morphological operations on input document image with broken text	12
3.2	Flow Chart for single skewed document images	13
3.3	Single skewed handwritten document image	14
3.4	Skew corrected output	14
4.1	Flow Chart for multi skewed multi handwritten or printed document images . . .	20
4.2	Multi Skewed handwritten text blocks	21
4.3	Skew Corrected document image	21
4.4	Skewed handwritten text in printed document image	22
4.5	Skew Corrected document image	22
4.6	Multiple skewed handwritten text in printed document	23
4.7	Multi Skew Corrected document image	24

Chapter 1

Introduction

1.1 Digital Image Processing in Documents

In document image processing, the paper documents are initially scanned and stored in the hard disk or any other external storage. It is easy to define document image processing as scanning, then storing, then, retrieving and then managing. The final output of processing of document image will be in compatible format, which makes documents easier and quicker to access. Document image processing comprises of a set of simple procedures and techniques, which are further used to work on the document images and convert them from information on pixels to a format that a computer can read.

Document image processing techniques will be more widely used, as all the computer-based entities and handwritten entities are electronically converted into documents. These e-documents are much easier to deal and we will be assured their existence for many years, only change is that they exist in cyber environment.

1.2 Problem Definition

- Line Segmentation in printed or handwritten documents.
- Word Segmentation in printed or handwritten documents.
- Skew Estimation in single skewed printed or handwritten documents.
- Multiple skew estimation in multi handwritten or printed documents.

1.3 Organization of thesis

The rest of the thesis is organized as follows.

Chapter 2:

Consists of techniques of image line and word segmentation assuming there is no skewness in the document.

Chapter 3:

Consists of skew estimation and correction in printed/handwritten documents assuming that the document.

Chapter 4:

Consists of a method to estimate multiple skews in a printed/handwritten document. This method also works if there are multiple handwritten texts with different skews in the document. This method can also work if there is a skewed handwritten text block on a printed document.

Chapter 2

Segmentation

2.1 Line Segmentation

2.1.1 Using Histogram Projection

Introduction to Histogram Projection:

Let H be height of image I and w be its width. Represent it as matrix without changing dimensions. We define the "projection profile"[11] of I as,

$$PP(k) = \sum_1^w I(x, y) \\ k = 1, 2, 3, \dots, H$$

Assuming that the text lines in the document image has same skew, applying histogram projection profiling on it gives a projection at each detected line. When skew angle is equal to zero, properties like frequency and amplitude of projection reach their maximum. While generating the histogram, we can use any filter that fits to our document image situation. The valleys that we can observe in the histogram signifies that there is high probability that a line exists.

Algorithm 1 :Algorithm for Line Segmentation using Histogram Projection

- 1: Read the image.
 - 2: Convert it into gray scale image and then to binary image.
 - 3: Generate a histogram.
 - 4: Smoothen the histogram using median filter.
 - 5: Find the minima in the histogram.
 - 6: These minima represent the valleys in the histogram which in turn indicates that they are the blank spaces between adjacent lines.
 - 7: Plot a line near each valley.
 - 8: END
-

Results

- Input (Handwritten):

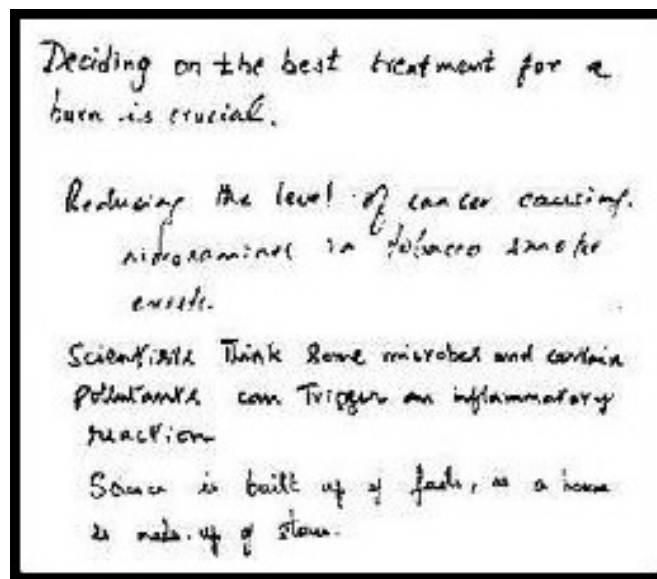


Figure 2.1: Input Handwritten document

- Output obtained using Histogram Projection:

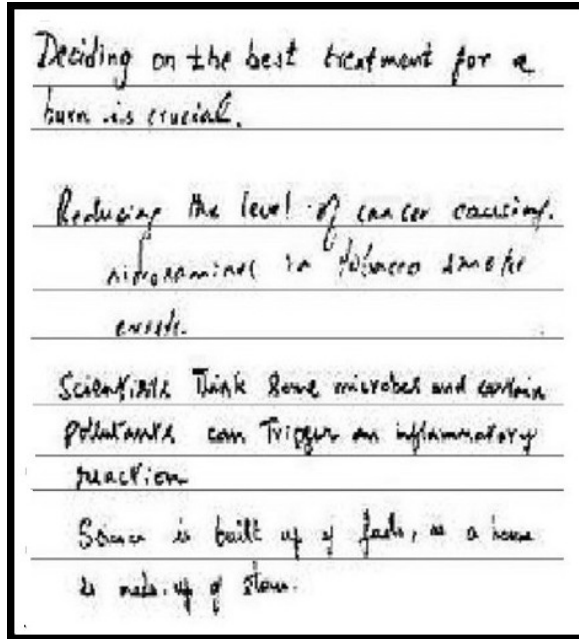


Figure 2.2: Horizontally segmented using Histogram Projection

2.1.2 Using Hough Transform

Introduction to Hough Transform:

Given a set of points in 2-D, find if a sub-set of these points, fall on a LINE. The solution is use Hough Transform [1, 10, 4]. The Hough transform is a line to point transformation from the Cartesian space to the Polar coordinate space. A line in the Cartesian coordinate space is described by the equation:

$$x\cos\theta + y\sin\theta = \rho$$

Each different line through the point (x^1, y^1) in (x, y) space corresponds to one of the points on the line in (m, c) space (termed as Hough space) [1, 10]. Two points in (x, y) space will now be represented by two different curves in (ρ, θ) space rather than straight lines. The intersection

corresponds to the parameters of the line in (x, y) space, formed by joining those two points.

Edge Detection (Using Canny Edge Detector):

Edges in images are areas with strong intensity contrasts - a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. The canny edge detector flow is given below.

It uses gaussian filter [4] to filter out noise in input image before detecting edges. Then, absolute gradient is calculated in both directions.

$$|Strength \rightarrow edge| = |G_x| + |G_y|$$

$$Theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

After that, non maximum suppression [4] is being used for tracing along edge directions. Now, using Hysteresis, it filters the extra noise.

Algorithm 2 Algorithm for Line Segmentation using Hough Transform

- 1: Input the image.
- 2: It has to be converted to gray scale if it is colored one in order to reduce the computational complexity. Apply morphological operations on it if necessary.
- 3: Use canny edge detector to detect the edges and adjust the parameter such that it will show all the edge in the image. The output image of the detector is binary one and it is not in continuous manner. So edge detection algorithm typically is followed by linking procedure to assemble edge pixel to meaningful edge.
- 4: Apply Hough Transform.

Syntax: $[h, \theta, \rho] = \text{hough}(f, d\theta, d\rho)$

Output: This function converts space of representation from xy-plane to $\rho\theta$ -plane. The output h is the Hough Transform matrix.

- 5: Peaks must be detected.

Syntax: $[r, c, h_{new}] = \text{hough_peaks}(h, num_peaks, threshold, n_hood)$

Output: r, c are row and column coordinates of the peaks identified. h_new is Hough Transform with suppressed peak neighborhood.

- 6: This step includes Hough Transform Line Detection and Linking.

Syntax: $[r, c] = \text{hough_pixels}(f, \theta, \rho, r_bin, c_bin)$

Output: This function computes the indices of rows, columns (r, c) for non-zero pixels in the image f that maps to Hough Transform bin, (r_bin, c_bin).

- 7: Group houghpixels into line segment.

Syntax: $Lines = \text{hough_lines}(f, \theta, \rho, rr, cc, fill_gap, min_length)$

Output: Final segmented image.

- 8: END
-

Results

- Input (Handwritten): Same input as given in the previous case.
- Output obtained using Hough Transform.

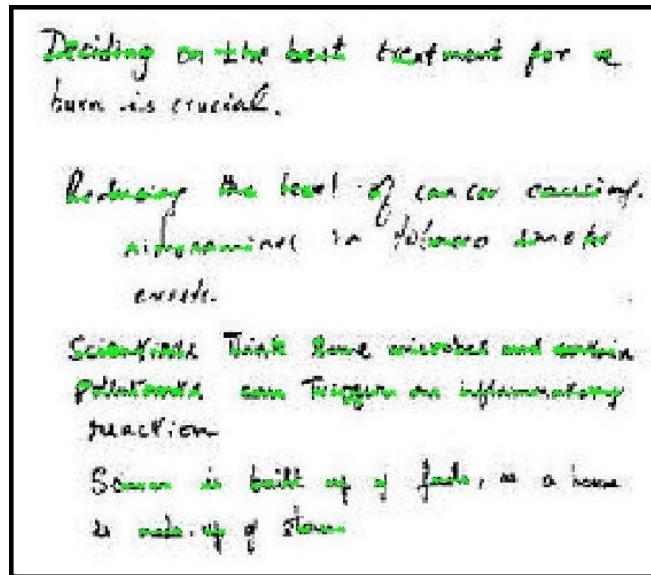


Figure 2.3: Segmented using Hough Transform (with low threshold value)

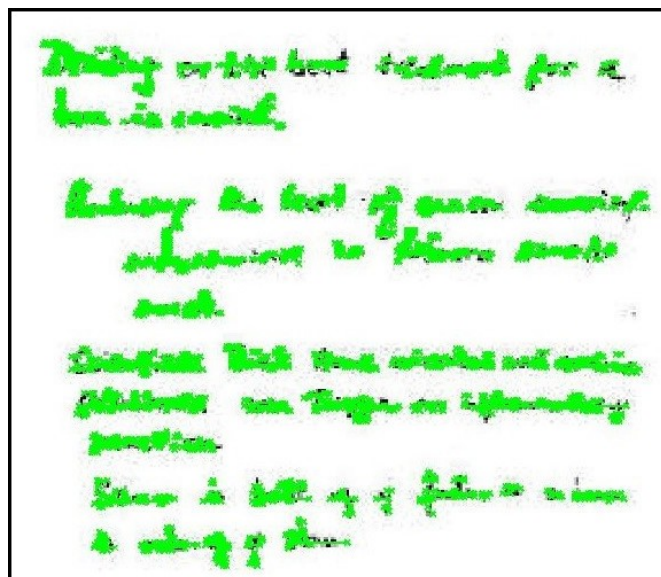


Figure 2.4: Segmented using Hough Transform (with high threshold value)

2.2 Word Segmentation

2.2.1 Connected Component Analysis

Connected components can be described as the groups that are formed by scanning any image and grouping together some pixels based on their connectivity. The scanning process is done from top to bottom and left to right. All these grouped pixels have similar intensities. The connected components are found using labeling.

If I represents the set of intensity values of a group (i.e. connected component) then,

- For binarized images, $I = 1$
- For gray scale images, $I = \text{range of values}$

Labelling of p

- If all four neighbors are 0, assign a new label to p .
- If only one neighbor has $I = 1$, assign its label to p .
- If more than one of the neighbors have $I = 1$, assign one of the labels to p , also make a note of the equivalences.

After image scan completion, the similar i.e. equivalent labels are sorted into equivalence classes. Now, allocate each label to a class. Make a second scan of the image and replace each label with its equivalent classes label.

Results

- Input (Handwritten)

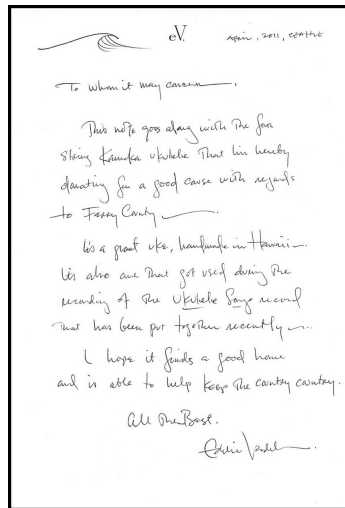


Figure 2.5: Input handwritten document

- Output - Word segmented using connected component analysis.

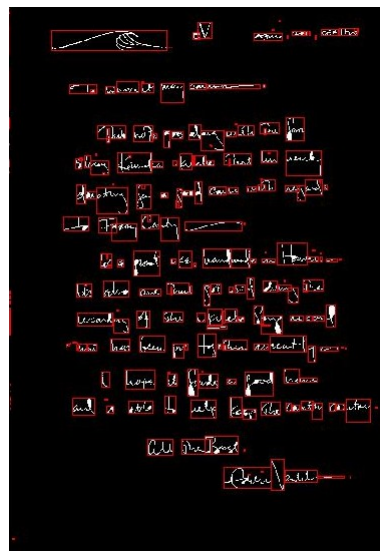


Figure 2.6: Word Segmented Output

Chapter 3

Skew Detection and Correction in single skewed printed or handwritten documents

The image of a printed or handwritten document may be rotated or skewed at an arbitrary angle because of how it was placed on the platen when it was scanned or because of a document feeder malfunction or the writing in handwritten documents may be skewed writing. This results in a skewed image. This represents a skew of only 5 degrees. In fact, a skew of as little as 0.1 degrees may be apparent to a human observer. Thus, a desirable function in a digital photocopier is the automatic detection and correction of skew. Ideally, an input such as that must produce skew corrected as output. A skew detection algorithm is given the image of a printed or handwritten document and it determines the angle (possibly zero degrees) by which it was skewed. It is assumed that there also exists a method for rotating the image to remove the skew (by using `imrotate()` in MATLAB).

A simple solution for skew detection is to determine the location of at least two corners of the original document and compute the skew angle from them. However, this can be error-prone because of non-linear distortions that occur when pages are not flat on the platen. Also, the entire scan surface may be obscured by the input document or the input may itself have been produced from a skewed original. In either case, deriving the skew angle from the corners or

edges of the page is problematic. In almost every case, the algorithms that are discussed assume that an input document contains some amount of text. Features are often extracted from the text portion of the image that allow the skew to be calculated. This is done because the text is usually structured into lines that are co-linear and aligned with the horizontal (or vertical) axis of the page. Thus, detecting the skew of the text lines provides the skew of the document.

3.1 Required Morphological Operations

If the input image is handwritten, it may have lot of noise (like broken texts). So initially apply series of appropriate morphological operations like dilation, erosion, skeletonization etc.

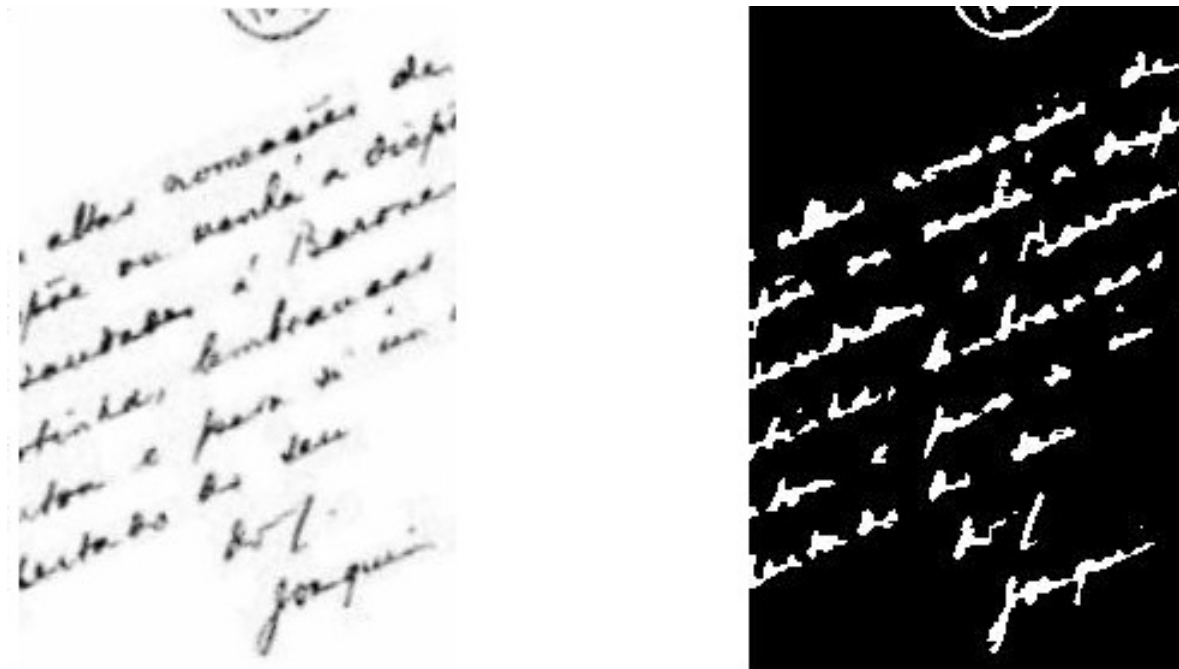


Figure 3.1: Image after applying morphological operations on input document image with broken text

3.2 Flow Chart of the method

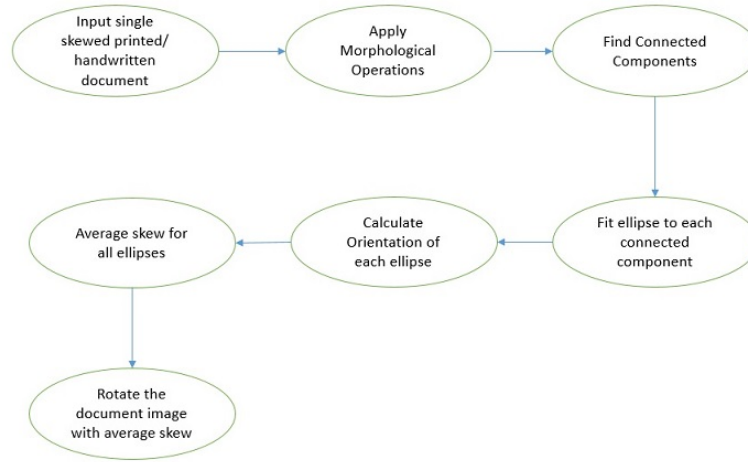


Figure 3.2: Flow Chart for single skewed document images

Description of steps:

After giving the printed or handwritten document as input, if there exists noise, eliminate it by subjecting the input to a series of appropriate morphological operations.

Segmenting the handwritten document is necessary in order to identify the different blocks written in different orientations. After segmenting, it would be easy to estimate the skew angles of multiple skewed blocks [6].

Next, identify the connected components in the image. The reason behind this step is that the skewness of a block depends on skewness of each connected component in it.

Now, fit a minimum circumscribing ellipse around each connected component throughout the image and calculate orientation of each ellipse.

Average all the orientations of ellipses which satisfy the threshold condition and rotate the input image with angle = "average skew angle".

3.3 Results

- Input



Figure 3.3: Single skewed handwritten document image

- Output

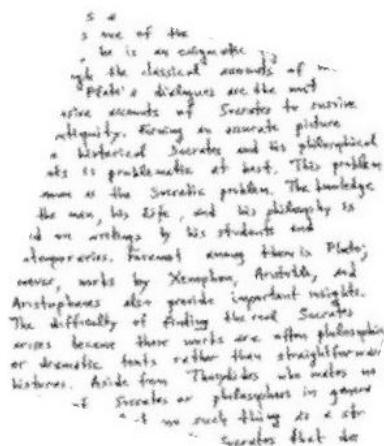


Figure 3.4: Skew corrected output

Chapter 4

Multi Skew Estimation and Correction in multi handwritten or printed documents

In indian scenario mostly, many of the office documents like forward notices and others are printed and have multi handwritten text over it with multiple skews. This poses a new challenge in "Document image analysis" field of research. In this view, there is a good method to estimate skew in multi skewed documents. This method mainly involves clustering techniques in skew estimation process.

4.1 Required Morphological Operations

If the input image is handwritten, it may have lot of noise (like broken texts). So initially apply series of appropriate morphological operations like dilation, erosion, skeletonization etc.

4.2 Clustering Analysis

Clustering is grouping a set of objects such that objects in a given group which is normally called as cluster are much more similar to one another than to those in other groups or clusters. Clustering is highly used in data mining, machine learning, statistical data analysis, information retrieval, pattern recognition.

Here mainly we use the following clustering algorithms.

- k-means Clustering
- Adaptive k-means Clustering[2]
- Spectral Clustering[9]

- **k-means Clustering**

Algorithm 3 :k-means Clustering Algorithm

- 1: Pick K cluster centers, either randomly or based on some heuristic.
 - 2: Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center.
 - 3: Re-compute the cluster centers by averaging all of the pixels in the cluster.
 - 4: Repeat steps 2 and 3 until convergence is attained (i.e. no pixels change clusters)
 - 5: END
-

- Spectral Clustering

Algorithm 4 :Spectral Clustering Algorithm

1: Given a set of coordinate points P in R^l that we should cluster into k clusters.

$$P = p_1, \dots, p_n$$

2: Construct the affinity matrix AM ,

$$AM \in R^{n \times n}$$

$$AM_{xy} = \|s_x - s_y\|$$

3: Construct diagonal matrix DM , whose (x, x) element is sum of AM 's x^{th} row, and construct matrix L .

$$L = (DM)^{-\frac{1}{2}}(AM)(DM)^{-\frac{1}{2}}$$

4: Calculate v_1, v_2, \dots, v_k the k largest eigenvectors of L and calculate the matrix V and stack all the present eigenvectors in columns.

$$V = [v_1 v_2 \dots v_k] R^{n \times k}$$

5: Calculate the matrix U from V by renormalizing each V 's rows to have a unit length.

$$U_{xy} = \frac{V_{xy}}{(\sum_y V_{xy}^2)^{1/2}}$$

6: Treat each row of U as point in R_k , cluster them into k clusters using "k means" or any other clustering algorithm that possibly attempts to minimize the distortion.

7: Assign the original point p_x to cluster y only if row x of the matrix U was assigned to cluster y .

8: END

• **Adaptive k-means Clustering**

Algorithm 5 :Adaptive k-means Clustering Algorithm

- 1: Calculate L matrix as of in spectral clustering method and take $z=2$.
- 2: Calculate z eigenvectors with highest eigenvalues and arrange them in U matrix.
- 3: Perform elongated k means with $z+1$ centers on U, initializing $(z+1)$ th mean in origin as shown below:

- First, initialize k number of centers c_1, c_2, \dots, c_k
- Repeat for each center (say c_i), calculate distance from it to all other points (say p)

If $c_i^T c_i > \epsilon$, that means center is not that close to origin.

$$\text{Distance} = (p, c_i) = (p - c_i)^T M (p - c_i)$$

$$M = \frac{1}{\lambda} (I_q - \frac{c_i c_i^T}{c_i^T c_i}) + \lambda (\frac{c_i c_i^T}{c_i^T c_i})$$

λ = amount of sharpness that controls elongation

If both origin and center are close enough, $c_i^T c_i < \epsilon$ then calculate distances as euclidian distance.

- Using this distance, assign each point "p" to its nearest center (i.e. centroid). Update location of each center by taking mean of data assigned to it.
 - Return to 2nd sub bullet and repeat until there is no change in centers.
- 4: If $(z + 1)^{th}$ cluster has any data points, then there must be at least an extra cluster, make $z++$ and go to 2nd step, else terminate the algorithm and output each cluster data and k (i.e. number of clusters).

5: END

4.3 Selection of Clustering Algorithms

If the document image consists of multiple handwritten text blocks as shown in input-1 in the next section, then k-means, spectral clustering [9], adaptive k-means [2] works just fine. But, in the first two clustering methods, we need to mention k.

Here, k can be selected manually or by a specified heuristic. This algorithm will converge, but the solution may not be optimal. The correctness of the solution depends on the initial set of clusters and the k.

The document image may contain of skewed handwritten text and a printed text block as shown in input-2 in the next section. This is a real life application where we can see such formal documents in passport offices, post offices and many other government offices. For this type, we can assume the value of k as 2 interpreting that the whole printed text is of single cluster and skewed handwritten text block is of another cluster. So k-means clustering works just fine.

Let's say the documents mentioned in above case contain two different skewed handwritten text blocks and on a printed one like in input-3. Here, we need to assume number of clusters as 3 unlike previous one. But, for like these cases, we observed that using k-means in this method which we used is giving abrupt results. Sometimes we are getting some wrong clusters as output. This problem is mainly due to randomized selection of centroids in the initial step of k-means clustering. This can be solved using spectral clustering [9] because it uses data based on eigen vectors. The output-3 is obtained using spectral clustering.

4.4 Flow Chart of the method

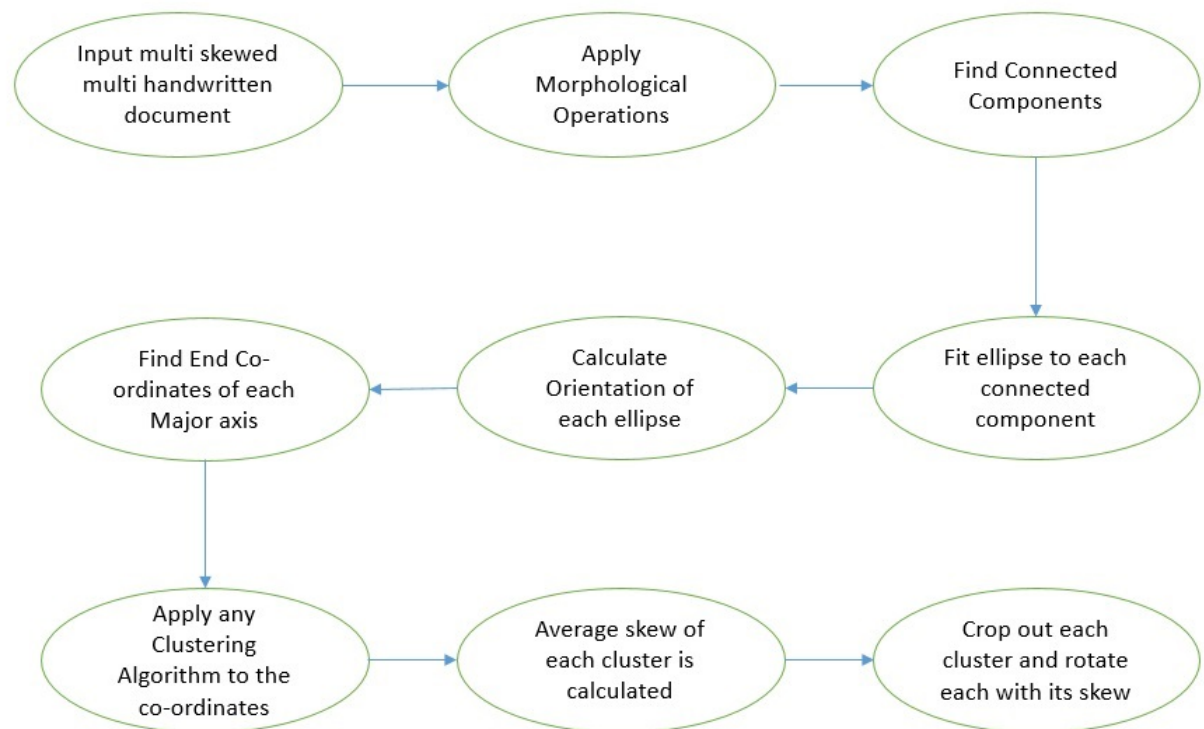


Figure 4.1: Flow Chart for multi skewed multi handwritten or printed document images

4.5 Results

- Input-1

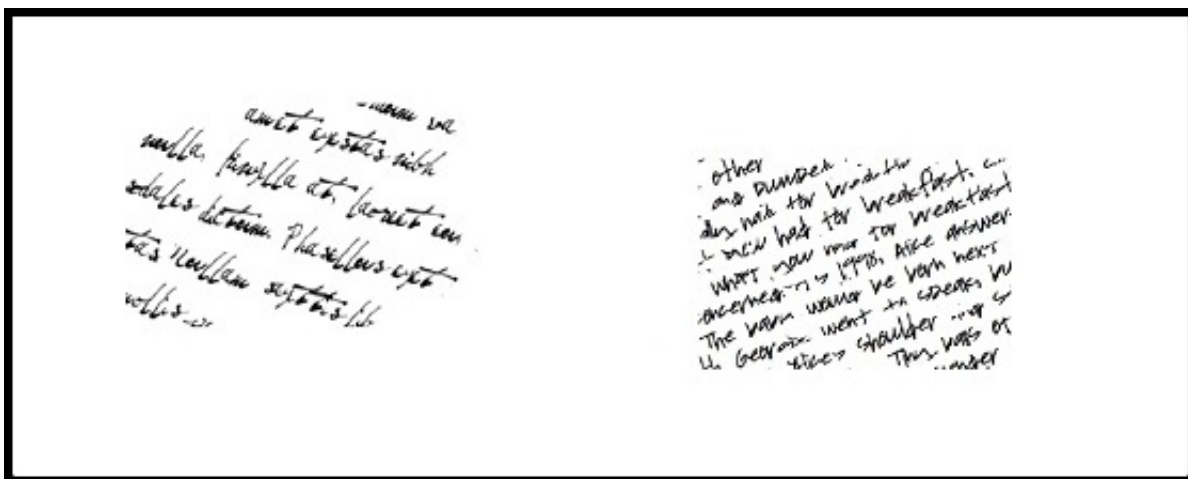


Figure 4.2: Multi Skewed handwritten text blocks

- Output-1

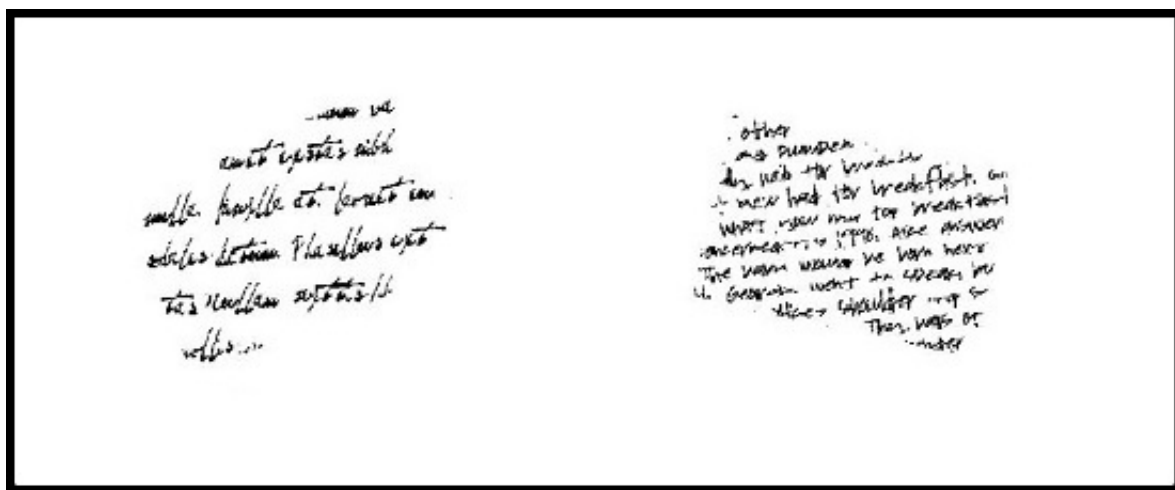


Figure 4.3: Skew Corrected document image

- Input-2

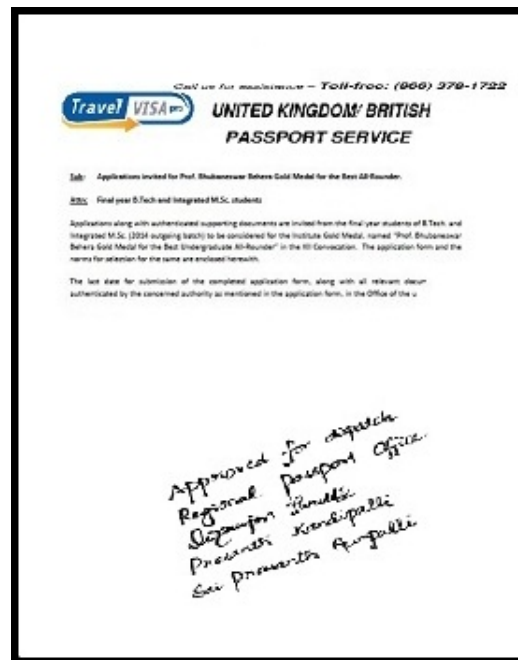


Figure 4.4: Skewed handwritten text in printed document image

- Output-2

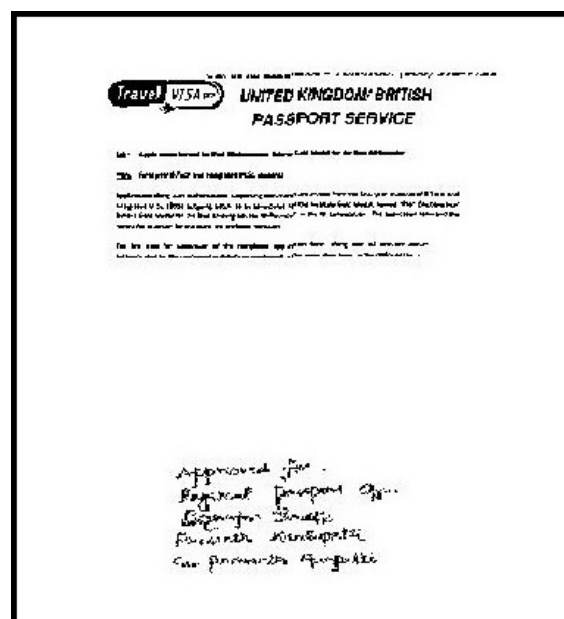


Figure 4.5: Skew Corrected document image

- Input-3

**Passport Application
Government of India, Ministry of External Affairs
Online Application Receipt**

No. /AC/2014/M/2549 Dated:

NOTICE

Sub: Applications invited for Prof. Bhubaneswar Behera Gold Medal for the Best All-Rounder.

Attn: Final year B.Tech and Integrated M.Sc. students

Applications along with authenticated supporting documents are invited from the final year students of B.Tech. and Integrated M.Sc. (2014 outgoing batch) to be considered for the Institute Gold Medal, named "Prof. Bhubaneswar Behera Gold Medal for the Best Undergraduate All-Rounder" in the XII Convocation. The application form and the norms for selection for the same are enclosed herewith.

The last date for submission of the completed application form, along with all relevant documents authenticated by the concerned authority as mentioned in the application form, in the Office of the u

*Approved for dispatch
Regional Passport Office
Deputy Commissioner
Prasanth Kandipalli
Sai Prasanth Gumpalli*

*Approved for dispatch
Regional Passport Office
Deputy Commissioner
Prasanth Kandipalli
Sai Prasanth Gumpalli*

Figure 4.6: Multiple skewed handwritten text in printed document

- Output-3

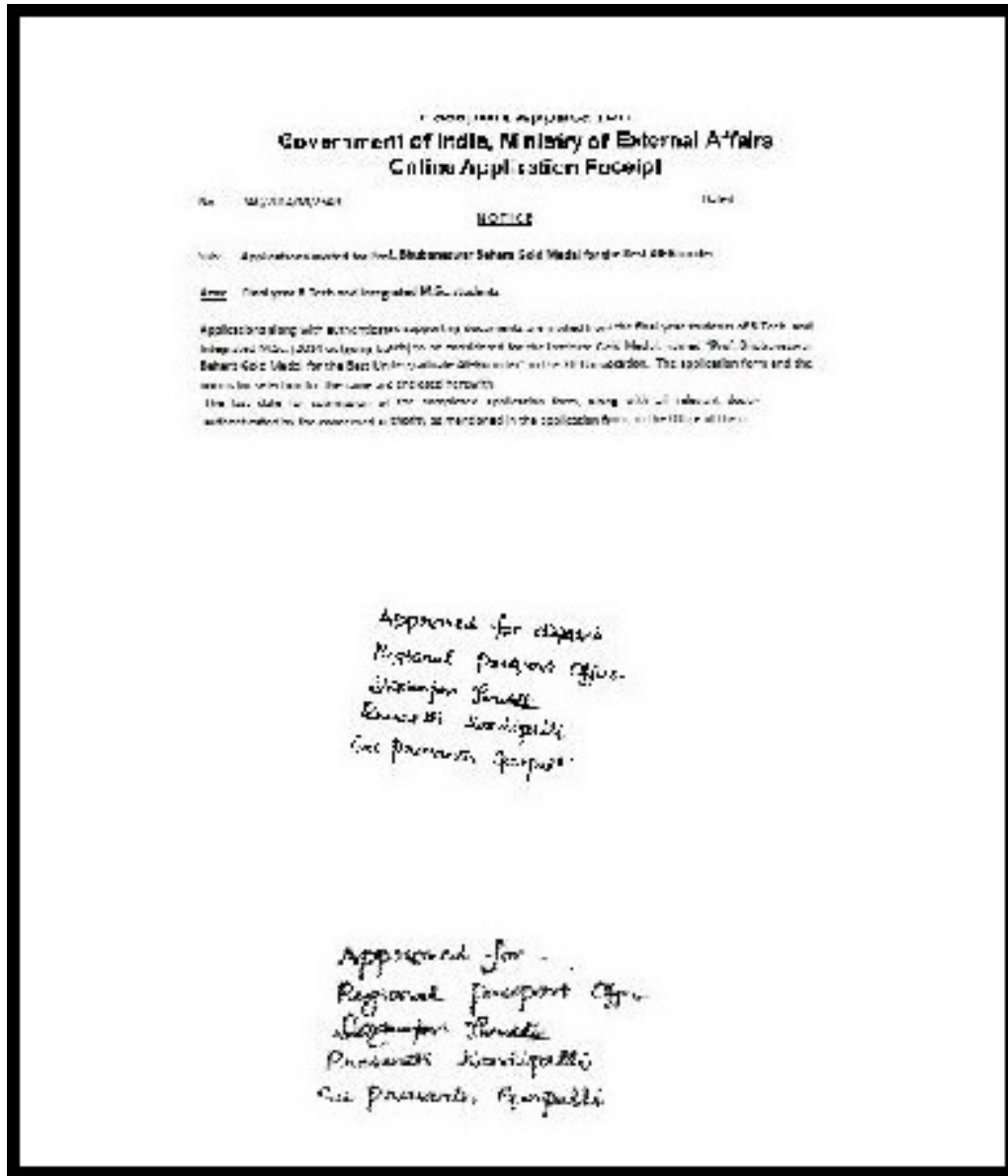


Figure 4.7: Multi Skew Corrected document image

Chapter 5

Conclusion and Future Work

The work in this thesis, primarily focuses on image segmentation and multi skew detection in both printed and handwritten document images. Summarization is done in this chapter. Section 5.1 lists the pros and cons. Section 5.2 gives a scope for developments that can be made further.

5.1 Achievements and Limitations

Handwritten Document Analysis [4] has always been a challenging area in image processing. Many algorithms have been developed for solving this problem. The algorithms implemented here for segmentation and skew detection works not only on printed or scanned document images but for also handwritten document images which creates an edge over other methodologies. However, there are certain limitations for some methods described here. For the method in chapter 3 and chapter 4, if the input handwritten is completely broken, a series of morphological operations [4] must be performed on the input image until we get a noiseless image. This is slightly time taking and also the morphological operations to be performed varies from document to document as they are handwritten. Also, in the method described in chapter 4, if the clusters are closely packed, it is becomes a little difficult for the program to identify clusters and in that case, output may not be as expected.

5.2 Further Developments

There is a scope to implement or modify the existing algorithm so that it can detect also detect shear skews in printed or handwritten document images.

Bibliography

- [1] G. Louloudis, K. Halatsis, B. Gatos, I. Pratikakis, "Hough Transform Mapping for Text Line Detection in Handwritten Documents", 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), Labaule, France, October 2006, pp. 515-520.
- [2] Sanguinetti G., Laidler J and Lawrence N. D.,: Automatic determination of the number of clusters using spectral algorithms. Proceedings of IEEE Machine Learning for Signal Processing, pp. 28-30. (2005)
- [3] N. Nandini, S. Murthy and G. H. Kumar, "Estimation of Skew Angle in Binary Document Images Using Hough Transform", World Academy of Science, Engineering and Technology, vol. 42, (2008), pp. 44-49.
- [4] Rafael C. Gonzalez and Richard E. Woods. Digital Image Processing. Prentice Hall, 2007. ISBN 0-13-168728-x.
- [5] Chandan Singh, Nitin Bhatia, Amandeep Kaur, "Hough transform based fast skew detection and accurate skew correction methods".
- [6] D. S. Guru¹, M. Ravi Kumar and S. Manjunath, Multiple Skew Estimation In Multilingual Handwritten Documents, IJCSI International Journal of Computer Science Issues, Vol.10, Issue 5, No 2, September 2013.
- [7] H.F. Jiang, C.C. Han, K.C. Fan, A fast approach to the detection and correction of skew documents, Pattern Recognition Lett. 18 (1997) 675686.
- [8] D.S. Le, G.R. Thomas, H. Wechsler, Automatic page orientation and skew angle detection for binary document images, Pattern Recognition 27 (1994) 13251344.

- [9] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, On spectral clustering: Analysis and an algorithm, in *Advances in Neural Information Processing Systems*, Thomas G. Dietterich, Sue Becker, and Zoubin Ghahramani, Eds., Cambridge, MA, 2002, vol. 14, MIT Press.
- [10] G. Louloudis, K. Halatsis, B. Gatos, I. Pratikakis, "A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents", 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France, October 2006, pp. 515-520.
- [11] Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti, "Text Line Segmentation Based on Morphology and Histogram Projection" Center of Informatics, Federal University of Pernambuco Recife, PE, Brazil